

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 0
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 3
- 9
- 0
- 1

## Overview

# Content

## None

## ApplicationControlMacro

## TransactionControlMacro

TResidency

# TLanguage

## TScheduleType

## TProgramType

## Boolean

**TMode**

## MessageType

## String

TSysID

TFPBuffer

## TCClass

TPriority

## TScheduleOption

## TParallelLimit

## TMaxRegions

## ProcessLimitType

UnsignedInteger

TSpaSize

TSegment

TResponse

## Classes

### *ApplicationControlMacro Class*

#### Introduction

TheApplicationControlMacro defines the program resource requirements for application programs that run under the control of the IMS DB/DC environment, as well as for applications that access databases through DBCTL. When combined with one or more TransactionControlMacros, the ApplicationControlMacro defines the scheduling and resource requirements for an application program. On the ApplicationControlMacro, one only describes programs that operate in message processing regions, Fast Path message-driven program regions, batch message processing regions, or CCTL threads. The ApplicationControlMacro is also used to describe application programs that operate in batch processing regions.

#### Base Classes

None

#### Attributes

**ResidencyOption:** TResidency

The positional parameter Resident specifies that the PSB associated with the application program is to be made resident during system initialization. Resident and DynamicOption (dynamic PSB option) are mutually exclusive. DynamicOption and ScheduleType=Parallel are also mutually exclusive.

Neither Resident nor DynamicOption is the default parameter. Rather, if neither Resident nor DynamicOption is selected on the ApplicationControlMacro statement, IMS system initialization causes a BLDL to be performed on the PSB associated with the application being defined. The PSB is not made resident (that is, loaded from the ACBLIB) until the application is scheduled.

**isFPath:** Boolean = False

Specifies whether the application is a Fast Path exclusive application program. If True, the ProgramType parameter is invalid and TransactionControlMacro.isWaitForInput=True is forced.

**FPathBuff rSize:** TFPBuffer

Determines the EMH buffer size required to run the transaction and overrides the EML execution parameter.

1

2

3

4

5

6

7

3

9

10

11

12

13

14

15

16

17

18

## Base Classes

## Attributes

TransactionCode: String

**Specifies the transaction code associated with this TransactionControlMacro statement.**

**isDCLogWriteAhead: Boolean**

Specifies whether IMS should perform log write-ahead for recoverable, nonresponse mode input messages and transactions output messages. If not specified in the TransactionControlMacro, the default is the DCLogWriteAhead parameter in the IMSCTRL macro. The DCLogWriteAhead parameter in the TransactionControlMacro overrides the IMSCTRL macro parameter for this transaction.

**toUpperCase: Boolean = True**

Specifies whether (True) or not (False) the input data is to be translated to uppercase before being presented to the processing program.

**Specifying True for VTAM terminals prevents the transmission of embedded device control characters.**

Specifying the attribute requires specifying the EditRoutine attribute.

**EditRoutine:** String

This attribute allows a user to a transaction input edit routine that edits messages prior to the program receiving the message. The specified edit routine (load module) must reside on the USERLIB data set prior to IMS system definition stage 2 execution. This routine cannot be the same as the one that is used on a LINEGRP or TYPE EDIT parameter.

**This attribute is invalid on a Fast Path exclusive transaction.**

**isFPPath:** Boolean = False

Specifies whether the transaction code is a potential candidate for Fast Path processing. Specifying True is effective only when specified on a TransactionControlMacro that follows an ApplicationControlMacro statement that does not specify isFPPath=True; otherwise, the operand is ignored.

Fast Path potential transactions must be processed by a user edit/routing exit to determine whether the transaction is actually to be processed by IMS Fast Path. If it is to be processed by IMS Fast Path, the edit/routing exit associates the transaction with a routing code. This routing code identifies which Fast Path application program is to process the transaction.

**FPathBufferSize:   TFPBuffer**

Determines the EMH buffer size required to run the transaction and overrides the EML execution parameter.

**isInquiry: Boolean = False**

Specifies whether the transaction is an inquiry transaction.

**isRecover: Boolean = True**

Specifies whether the transaction should be recovered during an IMS emergency or normal restart. If **isInquiry=False** is coded, **isRecover=False** cannot be coded.

For IMS Fast Path transactions, **isRecover** must be **True**.

If the SPA parameter is specified, the combination of **isInquiry=True** and **isRecover=False** must not be coded.

**MaxRegions: TMaxRegions = 0**

Limits the number of message processing program (MPP) regions that can be concurrently scheduled to process a transaction. When the number of MPP regions is not limited, one transaction can monopolize all available regions.

If the value is zero, no limit is imposed.

If **ApplicationControlMacro.ScheduleType=Serial** or **TransactionControlMacro.isSerial=True** is specified, the **MaxRegions** parameter must be omitted or zero.

**Mode: TMode = MultipleMessage**

Specifies that database buffers are to be written to direct access (flushed) upon each request for a new message (**SingleMessage**) by the processing program, or upon program termination (**MultipleMessage**). Conversational and **WaitForInput** transactions must be defined as **SingleMessage**. **SingleMessage** is forced for **WaitForInput** applications.

This operand is not appropriate if a transaction destined for processing in another (remote) system is being specified.

**MessageType: TSegment = MultipleSegment**

Specifies the type of transaction code (single or multiple segment).

The transaction code can be single segment (**SingleSegment**), or multiple segment (**MultipleSegment**). It specifies the time at which an incoming message is considered complete and available to be routed to an application program for subsequent processing.

The meaning of the values in detail:

**MultipleSegment**: Specifies that the incoming message is one segment in length. It is not eligible for scheduling to an application program until an end-of-message indication is received, or a complete message is created by MFS.

**SingleSegment**: Specifies that the incoming message is one segment in length. It becomes eligible for scheduling when the terminal operator indicates end-of-segment.

**MessageIsResponse: TResponse = NonResponse**

1 Specifies whether or not the communication line from the transaction was entered is to be held  
2 until a response is received.

3 The meaning of the values in detail:

4 NonResponse: Specifies that, for terminals specifying or defaulting to OPTIONS=TRANRESP,  
5 input should not stop after this transaction is entered.

6 Response: Specifies that, for terminals specifying or defaulting to OPTIONS=TRANRESP, no  
7 further messages are to be allowed after this transaction is entered until this transaction sends a  
8 response message back to the terminal. Response mode can be forced or negated by individual  
9 terminal definition.

10 MessageClass: TClass = 1

11 Specifies the class to which this transaction code is to be assigned.

12 ParallelLimit: TParallelLimit

13 Specifies the threshold value to be used when ApplicationControlMacro.ScheduleType=Parallel  
14 was specified in the preceding ApplicationControlMacro instruction. An additional region is  
15 scheduled whenever the current transaction enqueue count exceeds the ParallelLimit value  
16 multiplied by the number of regions currently scheduled for this transaction. If ParallelLimit is  
17 not specified, the default value of none is assumed, and IMS allows the transaction to be  
18 scheduled in only one region at a time.

19 ParallelLimit=0 indicates that any input message can cause a new region to be scheduled because  
20 the scheduling condition will always be met.

21 If ParallelLimit=0 is specified, the MaxRegions value should be specified to limit the number of  
regions that can be scheduled to process a particular transaction.

This operand is not appropriate if a transaction destined for processing in another (remote)  
system is being specified.

ProcessLimitCount: UnsignedInteger = 65535

Specifies the number of messages (count) of this transaction code a program can process in a  
single scheduling.

This field specifies the number of messages sent to the application program by the IMS control  
program for processing without reloading the application program. If 0 is coded, the maximum  
number of messages sent to the application is one and the application program is reloaded before  
receiving a subsequent message. The value 65535 means no limit is being set.

This operand is not appropriate if a transaction destined for processing in another (remote)  
system is being specified.

ProcessLimitSeconds: UnsignedInteger = 65535

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21

2  
3  
4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

20

21

1



1 4: Skip to the next class and attempt to schedule the highest priority transaction in that class.

2 This operand is not appropriate if a transaction destined for processing in another (remote)  
3 system is being specified.

4 **SegmentNumber: UnsignedInteger = 0**

5 Specifies the maximum number of application program output segments that will be allowed into  
6 the message queues per Get Unique (GU) call from the application program.

7 This operand is not appropriate if a transaction destined for processing in another (remote)  
8 system is being specified.

9 **SegmentSize: UnsignedInteger = 0**

10 Specifies the maximum number of bytes allowed in any one output segment.

11 This operand is not appropriate if a transaction destined for processing in another (remote)  
12 system is being specified.

13 **isSerial: Boolean = False**

14 Specifies whether the processing of messages for a given transaction is forced to be serial.

15 If True is specified, ParallelLimit and MaxRegions must equal zero (they can be omitted). If they  
16 are not, False is assumed.

17 This operand is not appropriate if a transaction destined for processing in another (remote)  
18 system is being specified.

19 **SPASize: TSpaSize**

20 Defines by inclusion that this transaction is a conversational transaction.

21 The value specifies the size of the conversational scratchpad area (SPA).

**SPAIsTruncated: Boolean = True**

Specifies whether the truncated data option is used with the scratchpad area. .

**SysIDRemote: TSysID**

Specifies in the multiple IMS system configuration, the system identification (SysID) of the  
remote system (that system on which the application executes).

If the ApplicationControlMacro.SysID parameters are specified, the  
TransactionControlMacro.SysID parameters don't need to be specified. If both SysID parameters  
are specified, the ApplicationControlMacro.SysID parameters are ignored.

**SysIDLocal: TSysID**

Specifies in the multiple IMS system configuration, the system identification (SysID) of the local  
system (the originating system to which the response are returned).

If the `ApplicationControlMacro.SysID` parameters are specified, the `TransactionControlMacro.SysID` parameters don't need to be specified. If both `SysID` parameters are specified, the `ApplicationControlMacro.SysID` parameters are ignored.

**isWaitForInput:** Boolean

Specifies whether (True) or not (False) this is a wait-for-input transaction. A message processing or batch processing application program that processes WaitForInput transactions is scheduled and invoked normally. If the transaction to be processed is defined as WaitForInput, the program is allowed to remain in main storage after it has processed the available input messages. The QC status code (no more messages) is returned to the program if the ProcessLimitCount value is reached.

**Mode=SingleMessage** will be forced, if **isWaitForInput=True** is coded.

This operand is not appropriate if a transaction destined for processing in another (remote) system is being specified.

## Inherited Attributes

## Associations

## APPLCTN with ApplicationControlMacro

### *TResidency enumeration*

### Attributes

**Resident**

## DynamicOption

### *TLanguage enumeration*

### Attributes

## Assembler

# COBOL

**PL/1**

**PLA**

## Pascal

### *TScheduleType enumeration*

## Attributes

**Serial**

## Parallel

### *TProgramType* enumeration

### Attributes

1	<b>TeleProcessing</b>
2	<b>Batch</b>
3	<b><i>Boolean enumeration</i></b>
4	<b>Attributes</b>
5	<b>False</b>
6	<b>True</b>
7	<b><i>TMode enumeration</i></b>
8	<b>Attributes</b>
9	<b>MultipleMessage</b>
10	<b>SingleMessage</b>
11	<b><i>MessageType enumeration</i></b>
12	<b>Attributes</b>
13	<b>MULTSEG, NONRESPONSE</b>
14	<b>SNGLSEG, RESPONSE</b>
15	<b><i>String primitive</i></b>
16	<b>length: 1..8</b>
17	<b>valid characters: A through Z, #, \$, @, 0 through 9</b>
18	<b><i>TSysID primitive</i></b>
19	<b>Valid values: 1..2036</b>
20	<b><i>TFPBuffer primitive</i></b>
21	<b>Valid values: 12..30720</b>
	<b><i>TClass primitive</i></b>
	<b>valid values: 1..255</b>
	<b><i>TPriority primitive</i></b>
	<b>valid values: 0..14</b>
	<b><i>TScheduleOption primitive</i></b>
	<b>valid values: 1..4</b>
	<b><i>TParallelLimit primitive</i></b>
	<b>valid values: 0..32767</b>

- 1 *TMaxRegions primitive*  
valid values: 0..255
- 2 *ProcessLimitType primitive*  
valid values: 1..65535
- 3 *UnsignedInteger primitive*  
valid values: 0..65535
- 4 *TSpaSize primitive*  
valid values: 16..32767
- 5 *TSegment enumeration*  
Attributes
  - 6 MultipleSegment
  - 7 SingleSegment
- 8 *TResponse enumeration*  
Attributes
  - 9 Response
  - 10 NonResponse
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21